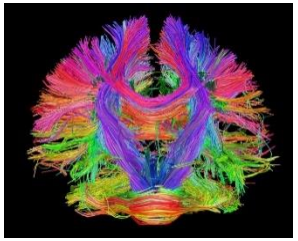



CS-570
Statistical Signal Processing

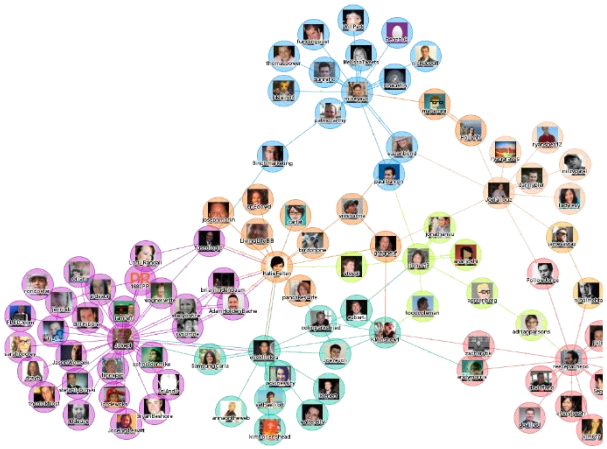
Lecture 14: Graph Signal Processing

Spring Semester 2019

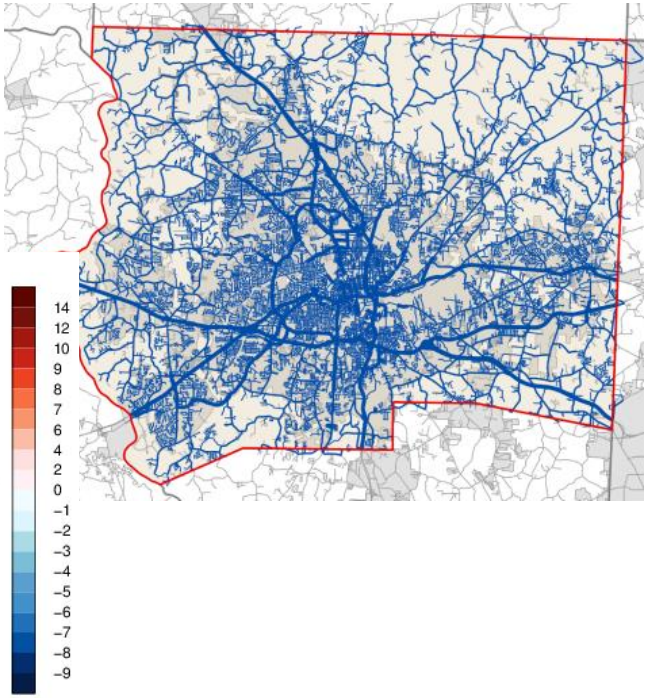
Grigorios Tsagkatakis

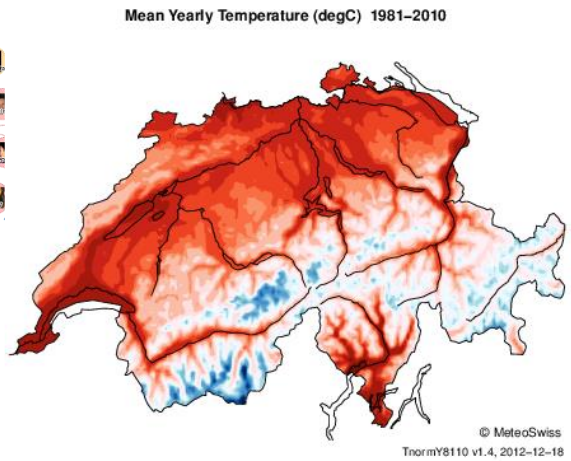
Types of signals on graphs

Social networks



Electrical networks





Environmental monitoring

Modeling signals on graphs

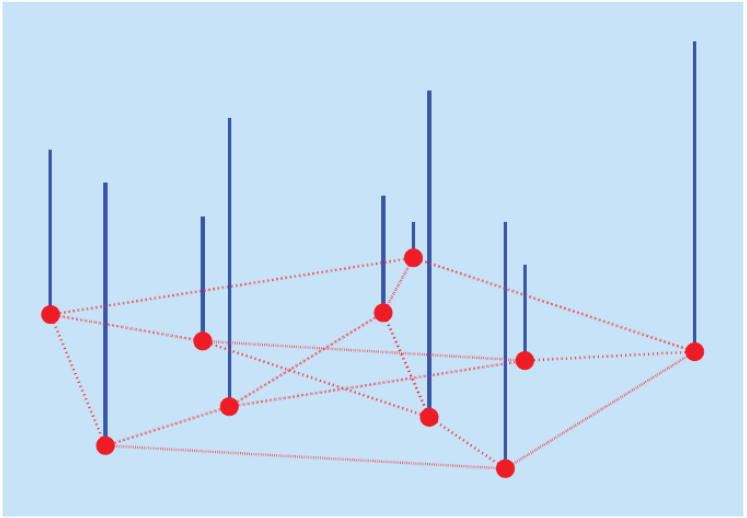
Edge connectivity/weight <-> similarity between vertices.

Known connectivity

- Social media
- Sensor networks
- Traffic networks
- 3D point clouds

Unknown connectivity

- Neuronal networks
- Internet/WWW
- Gene regulatory networks



The height of each blue bar represents the signal value at the vertex.

Why analyze signals on graphs

- Epidemiological data describing the spread of disease
- Census data describing human migration patterns
- Logistics data describing inventories of trade goods
- Anatomical connectivity of distinct functional regions of the cerebral cortex
- Cluster different genes based on their phenotype/participation in metabolism
- Classify human activity from depth sensors

Why

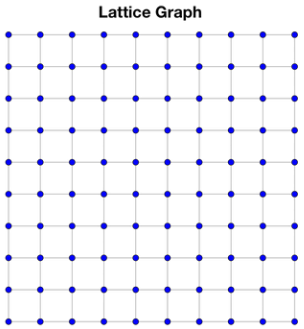
Common data processing tasks:

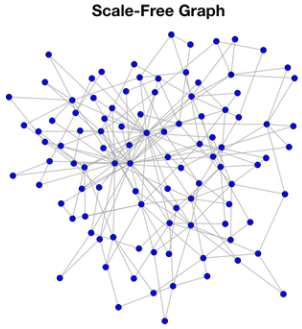
- Filtering
- Denoising
- Inpainting
- Compression

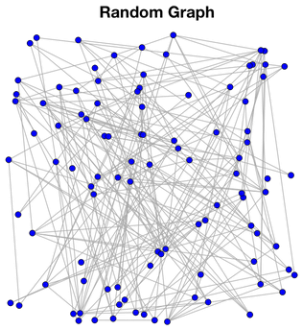
Challenges

- What is translation over a grah ?
- What is downsampling over a graph ?

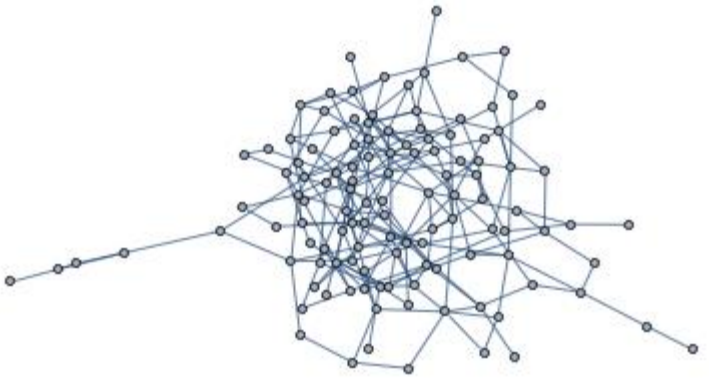
Types of graphs

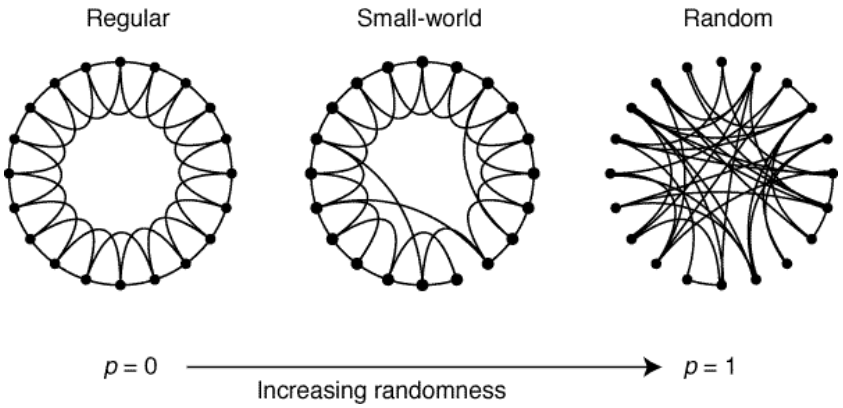


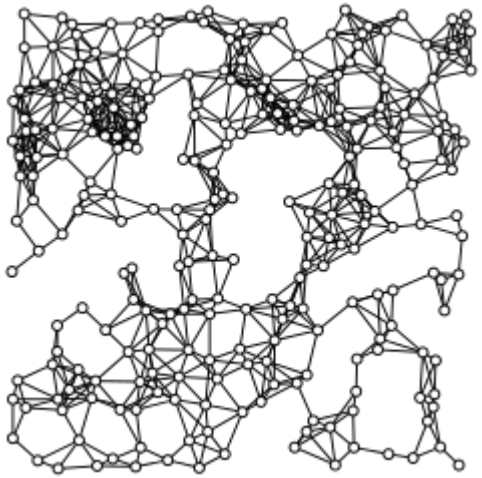




- Erdos-Renyi graphs,
- ring graphs,
- Random geometric graphs,
- small-world graphs,
- power-law graphs,
- nearest-neighbor graphs,
- scale-free graphs







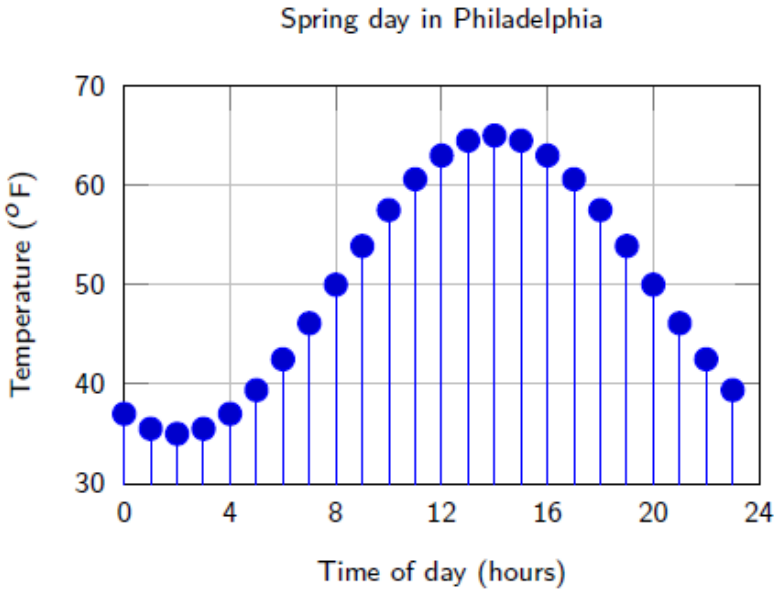
Regular graph structures

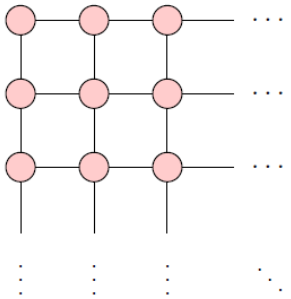
1D Time-series

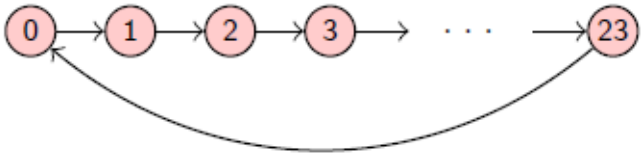
- Nodes <-> time instances
- Edges are unweighted and directed

2D images

- Nodes <-> pixel
- Edges <-> similarity







Signals on Graphs

Graphs: generic data representation forms encoding the geometric structures of data

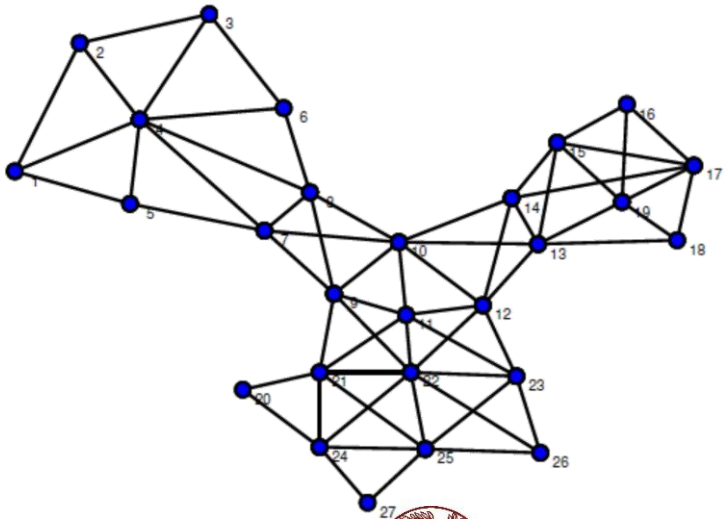
Applications: social networks, energy distribution networks, transportation network, **wireless sensor network**, and neuronal networks.

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$$

vertices edges weighted adjacency matrix

weights: distance /similarity/relationship

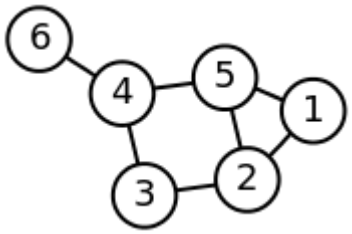
$$W_{i,j} = \begin{cases} \exp\left(-\dfrac{[\mathrm{dist}(i,j)]^2}{2\theta^2}\right) & \text{if } \mathrm{dist}(i,j) \le \kappa \\ 0 & \text{otherwise,} \end{cases}$$



Assumptions
1. Undirected graphs without self loops.
2. Scalar sample values

Modeling the graph

Undirected graph



Degree matrix: **D**

$$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Adjacency matrix: **A**

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Laplacian matrix: **L**

$$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

The **adjacency matrix** is a matrix, $\mathbf{A}$, such that $\mathbf{A}_{ij} = w_{ij}$.

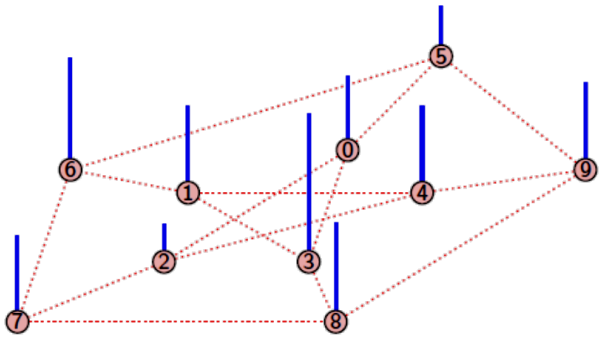
if the graph is undirected, $w_{ij} = w_{ji}$, and $\mathbf{A}$ is symmetric

The **degree matrix** of G is a diagonal matrix, $\mathbf{D}$,
with entries $(\mathbf{D})_{ii} = \sum_{j=1}^{N} \mathbf{A}_{ij}$ and $(\mathbf{D})_{ij} = 0$ for $i \neq j$,

The **combinatorial graph Laplacian** defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$,
and the **symmetric normalized Laplacian** $\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.

Signals on Graphs

Graph signal f in R^N, where |V|=N



$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}$$

Graph Laplacian $\mathcal{L} := \mathrm{D} - \mathrm{W}$, D: diagonal with sums of weights

W: weight matrix

Normalized Graph Laplacian $\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$

The z transform

Consider N samples of a signal s_n, $n = 0, 1, \cdots, N-1$
of finite number N of samples and to filters
with finite impulse response (FIR filters)

The z-transform $s(z)$ of the time signal
$s = \{s_n : n = 0, 1, \cdots, N-1\}$ organizes its samples s_n
into an ordered set of time samples, where sample s_n
at time n precedes s_{n+1} at time $n+1$
and succeeds s_{n-1} at time $n-1$.

In other words, the signal is given by the
N-tuple $s = (s_0, s_1, \cdots, s_{N-1})$.

The z transform

This representation is achieved by using a formal variable z^{-1} called the shift (or delay), so that the signal of N-samples is represented by

$$s(z) = \sum_{n=0}^{N-1} s_n z^{-n}.$$

The DFT transform

The discrete Fourier transform (DFT) of the signal s is $\widehat{s} = \{\widehat{s}_k : k = 0, \cdots, N-1\}$ given by

$$\widehat{s}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} s_n e^{-j\frac{2\pi}{N}kn}.$$

The $\widehat{s}_k$ are the Fourier coefficients of the signal.

The discrete frequencies are $\Omega_k = \frac{2\pi k}{N}$, $k = 0, 1, \cdots, N-1$, and the N signals $(x_k[n])$

$$\left\{ x_k[n] = \frac{1}{\sqrt{N}} e^{-j\frac{2\pi}{N}kn} : n = 0, 1, \cdots, N-1 \right\}_{k=0}^{N-1}$$

are the spectral components.

The DFT transform

The signal is recovered from its Fourier coefficients by the inverse DFT:

$$s_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \widehat{s}_k e^{j\frac{2\pi}{N}kn}, s = 0, 1, \cdots, N-1.$$

FIR filters

An FIR filter is also represented by a polynomial in z^{-1}

$$h(z) = \sum_{n=0}^{N-1} h_n z^{-n},$$

so that the output s_{out} of filter h applied to signal s_{in} is:

$$s_{\text{out}}(z) = h(z) \cdot s_{\text{in}}(z).$$

Defining the shift or delay filter $h_{\text{shift}}(z) = z^{-1}$,
and applying it to a signal $s_{\text{in}} = (s_0, s_1, \cdots, s_{N-1})$ produces:

$$s_{\text{out}} = h_{\text{shift}} \cdot s_{\text{in}} = (s_{N-1}, s_0, s_1, \cdots, s_{N-2}).$$

Shift invariance

The series combination of filters is commutative,
a filter commutes with the shift filter—delaying the input signal and then filtering the delayed input signal
leads to the same signal as first filtering the input signal s_{in} and then delaying the filtered output.

$$z^{-1} \cdot h(z) = h(z) \cdot z^{-1}.$$

Writing the signal $s = (s_0, s_1, \cdots, s_{N-1})$ as the vector

$$\mathbf{s} = [s_0\, s_1\, \cdots\, s_{N-1}]^\top \in \mathbb{C}^N,$$

and a filter h as a matrix $\mathbf{H}$, filtering can be written as

$$\mathbf{s}_{out} = \mathbf{H} \cdot \mathbf{s}_{in}$$

Shift filtering operation

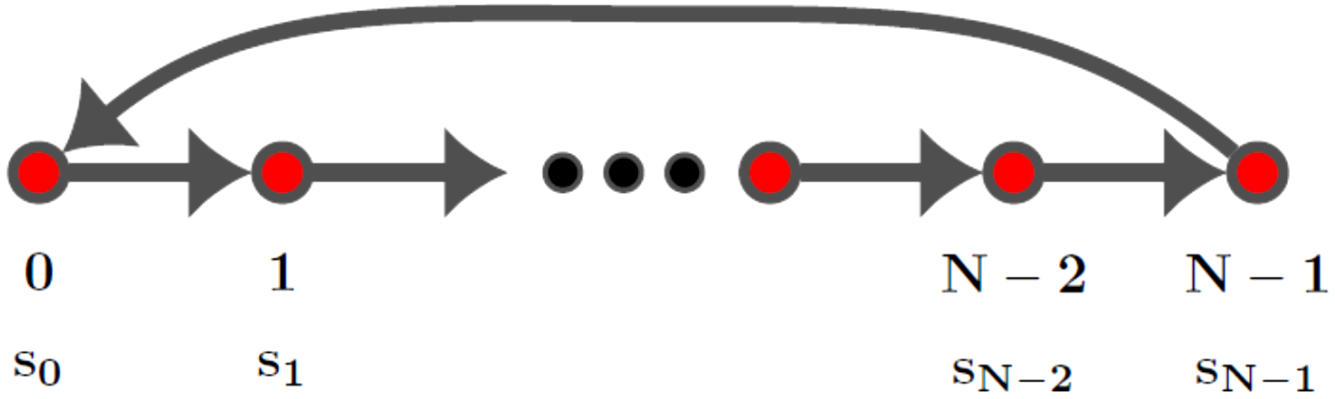
The shift filtering operation corresponds to multiplication by a circulant matrix $\mathbf{A}_c$

$$[s_{N-1}\, s_0\, \cdots\, s_{N-2}]^\top = \mathbf{A}_c \cdot [s_0\, s_1\, \cdots\, s_{N-1}]^\top ,$$

given by the cyclic shift

$$\mathbf{A}_c = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

Shift operation in 1D time graphs



The 0-1 shift matrix $\mathbf{A_c}$ as the adjacency matrix of a graph. Labeling the rows and columns of $\mathbf{A_c}$ from 0 to $N-1$, define the graph $G_c = (V, E)$ with node set $V = \{0, 1, \cdots, N-1\}$.

Shift invariance

A filter represented by $\mathbf{H}$ will be shift invariant
if it commutes with the shift: $\mathbf{AH} = \mathbf{HA}$

In GSP, filters are defined as matrices and the eigensignals of h are the eigenvectors of the corresponding $\mathbf{H}$.

Under certain conditions, every filter commuting with $\mathbf{A}$ is a polynomial in $\mathbf{A}$, i.e. $\mathbf{H} = h(\mathbf{A})$

Then $\mathbf{H} = h(\mathbf{A}) = \mathbf{V}h(\mathbf{\Lambda})\mathbf{V}^{-1}$, where $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$

Graph Fourier Transform (GFT)

The graph Fourier transform of graph signal $\mathbf{s}$ is given by the graph Fourier *analysis* decomposition

$$\widehat{\mathbf{s}} = \mathbf{F}\mathbf{s} = \mathbf{V}^{-1}\mathbf{s} = [f_0\mathbf{s} \cdots f_{N-1}\mathbf{s}]^\top$$

The graph Fourier coefficients or graph spectral coefficients of signal $\mathbf{s}$ are computed using the inner product as $\widehat{s}(\lambda_k) = \widehat{s}_k = f_k\mathbf{s} = \langle f_k^H, \mathbf{s} \rangle$.

The inverse GFT is given by

$$\mathbf{s} = \mathbf{F}^{-1}\widehat{\mathbf{s}} = \mathbf{V}\widehat{\mathbf{s}}$$

Filters on graphs

- **Theorem:** Let **A** be the graph adjacency matrix and assume that its characteristic and minimal polynomials are equal: $p_{\mathbf{A}}(x) = m_{\mathbf{A}}(x)$. Then, a graph filter **H** is linear and shift invariant if and only if (iff) **H** is a *polynomial* in the graph shift **A**, i.e., iff there exists a polynomial

$$h(x) = h_0 + h_1 x + \ldots + h_L x^L$$

with possibly complex coefficients $h_\ell \in \mathbb{C}$, such that:

$$\mathbf{H} = h(\mathbf{A}) = h_0 \mathbf{1} + h_1 \mathbf{A} + \ldots + h_L \mathbf{A}^L.$$

- The coefficients h_ℓ in the polynomial $h(x)$ are called the graph filter *taps*.

Filtering in Graph Frequency domain

Given the adjacency matric $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$

The graph filter can be expressed as

$$\begin{aligned}\mathbf{H} &= h(\mathbf{A}) \\ &= h\left(\mathbf{V}\mathbf{A}\mathbf{V}^{-1}\right) \\ &= \sum_{m=0}^{M-1} h_m \left(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}\right)^m \\ &= \mathbf{V} h\left(\mathbf{\Lambda}\right) \mathbf{V}^{-1},\end{aligned}$$

where $h\left(\mathbf{\Lambda}\right)$ is the diagonal matrix

$$h\left(\mathbf{\Lambda}\right) = \text{diag}\left[h\left(\lambda_0\right) \cdots h\left(\lambda_{N-1}\right)\right].$$

Filtering in Graph Frequency domain

The output of $\mathbf{s}_{in}$ to filter h is successively

$$
\begin{aligned}
\boldsymbol{s}_{\text{out}} &= \mathbf{H} \cdot \boldsymbol{s}_{\text{in}} \\
&= \mathbf{V} h(\boldsymbol{\Lambda}) \underbrace{\left(\mathbf{V}^{-1} \boldsymbol{s}_{\text{in}}\right)}_{\text{Fourier transf.}} \\
&= \mathbf{V} \underbrace{\operatorname{diag}\left[h(\lambda_0) \cdots h(\lambda_{N-1})\right] \widehat{\boldsymbol{s}}_{\text{in}}}_{\text{Filtering in graph Fourier space}} \\
&= \underbrace{\mathbf{V} \left[h(\lambda_0)\, \widehat{\boldsymbol{s}}_{\text{in}_0} \cdots h(\lambda_{N-1})\, \widehat{\boldsymbol{s}}_{\text{in}_{N-1}}\right]^{\top}}_{\text{Inverse Fourier transf.}}.
\end{aligned}
$$

Using the Graph Laplacian

A frequency representation can be similarly built
on top of the Laplacian matrix of an undirected graph.

Since this matrix is positive semidefinite,
all the eigenvalues are real and non-negative,
and a full set of orthogonal eigenvectors can be obtained,
so that we can write

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\top}$$

with $\mathbf{U}$ the GFT matrix, which is real and orthogonal.

Because the eigenvalues are real, they provide a natural way
to order the GFT basis vectors in terms of frequency.

GFT cont'd

- Graph Fourier Transform

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T = \sum_{i=1}^{N} \lambda_i \mathbf{u}_i \mathbf{u}_i^T,$$

- GFT : projection onto the eigenvectors of the graph Laplacian

$$\hat{\mathbf{x}} = \mathbf{U}^T \mathbf{x}$$

- Inverse GFT:

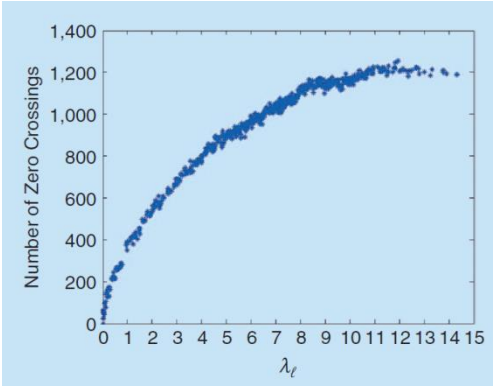
$$\mathbf{x} = \mathbf{U}\,\hat{\mathbf{x}}$$

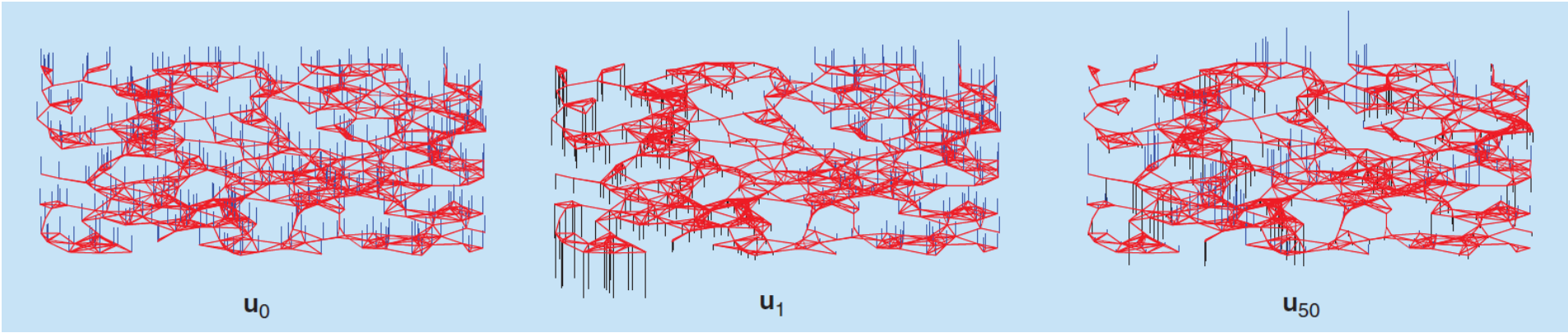
- The graph Laplacian eigenvectors associated with low frequencies vary slowly across the graph
- The eigenvectors associated with larger eigenvalues oscillate more rapidly

Graph Laplacian

Spectral properties $\mathcal{L}\mathbf{u}_\ell = \lambda_\ell \mathbf{u}_\ell$

- Laplacian is Positive Semi-definite matrix
- Eigenvalues: $0=\lambda_1(L) \leq \lambda_2(L) \leq \ldots \leq \lambda_{N-1}(L)$
- Eigen-pair system $\{\lambda_k, \mathbf{u}_k\}$ provides Fourier-like interpretation (GFT)





$L = \chi \Lambda \chi^T$

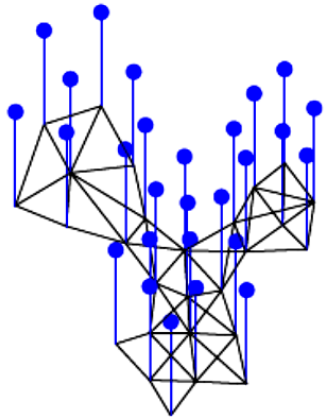
Low frequency → High frequency

$\chi_0^T L \chi_0 = \lambda_0 = 0$

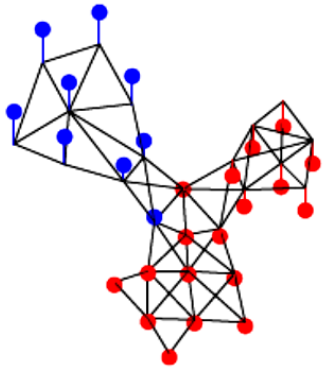
$\chi_{50}^T L \chi_{50} = \lambda_{50}$

Eigenvectors of Graph Laplacian

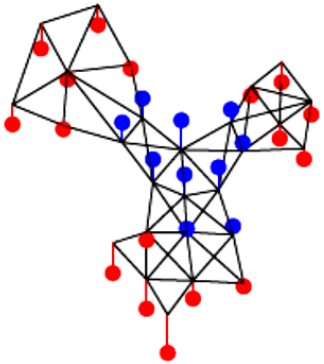
(a) $\lambda = 0.00$



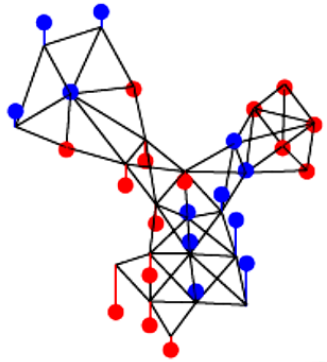
(b) $\lambda = 0.04$



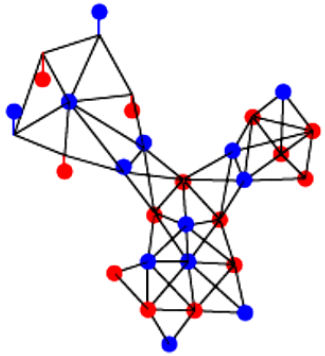
(c) $\lambda = 0.20$



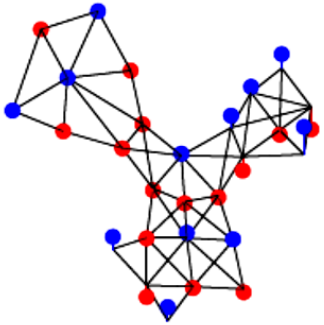
(d) $\lambda = 0.40$



(e) $\lambda = 1.20$



(f) $\lambda = 1.49$



Importance

For connected graphs, the Laplacian eigenvector $\mathbf{u}_0$ associated with the eigenvalue 0 is constant and equal to $\frac{1}{\sqrt{N}}$ at each vertex.

The graph Laplacian eigenvectors associated with low frequencies vary slowly across the graph.

If two vertices are connected by an edge with a large weight, the values of the eigenvector at those locations are similar. The eigenvectors associated with larger eigenvalues oscillate more rapidly and are more likely to have dissimilar values on vertices connected by an edge with high weight.

Example

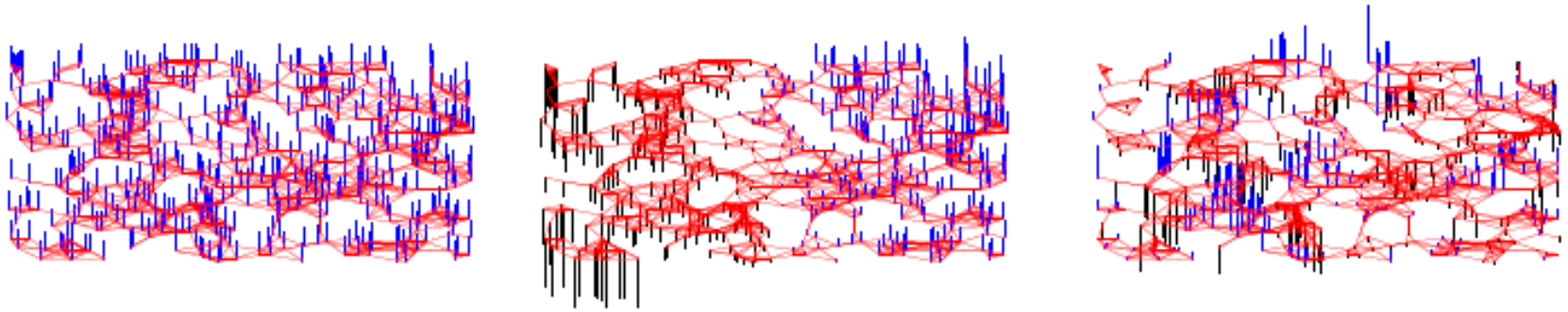
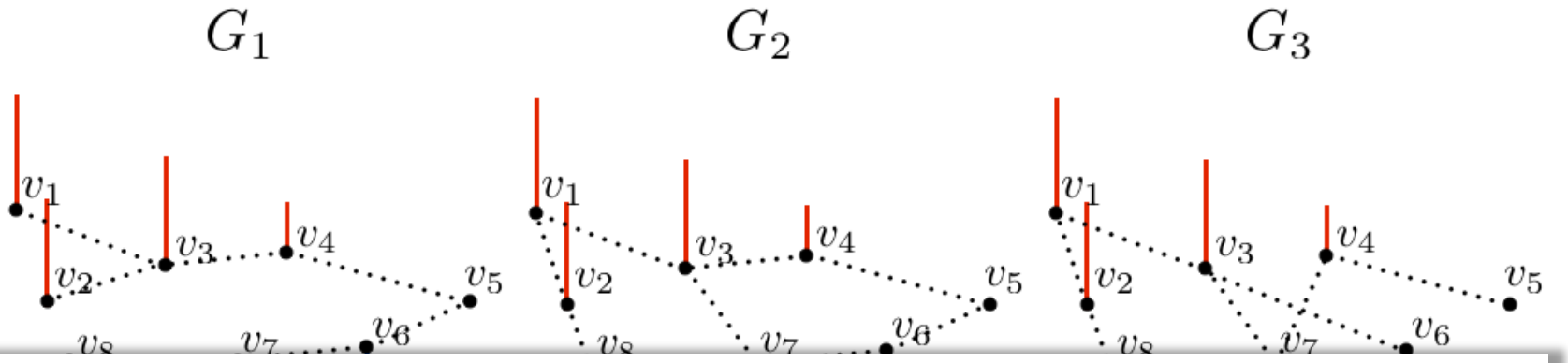


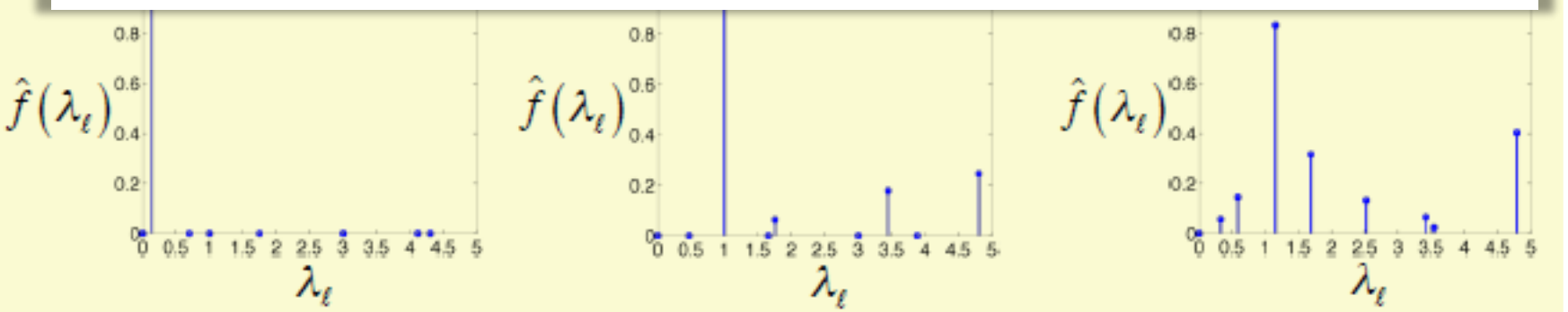
Fig. 2. Three graph Laplacian eigenvectors of a random sensor network graph. The signals' component values are represented by the blue (positive) and black (negative) bars coming out of the vertices. Note that $\mathbf{u}_{50}$ contains many more zero crossings than the constant eigenvector $\mathbf{u}_0$ and the smooth *Fiedler vector* $\mathbf{u}_1$.

Estimating the underlying graph



- One signal <-> many different graphs
- Only 1 leads to a smooth graph signal.

➢ Only G_1 favors smoothness of the resulting graph signal.



Graph smoothness

Graph based approximation $\hat{f}(\lambda_\ell) := \langle \mathrm{f}, \mathrm{u}_\ell \rangle = \sum_{i=1}^{N} f(i)\, u_\ell^*(i).$

Smoothness w.r.t. graph $\| \mathrm{f} \|_{\mathcal{L}} := \| \mathcal{L}^{\frac{1}{2}} \mathrm{f} \|_2 = \sqrt{\mathrm{f}^{\mathrm{T}} \mathcal{L} \mathrm{f}} = \sqrt{S_2(\mathrm{f})}.$

Graph spectral filtering (regularization) $\min_{\mathrm{f}} \{ \| \mathrm{f} - \mathrm{y} \|_2^2 + \gamma S_p(\mathrm{f}) \},$

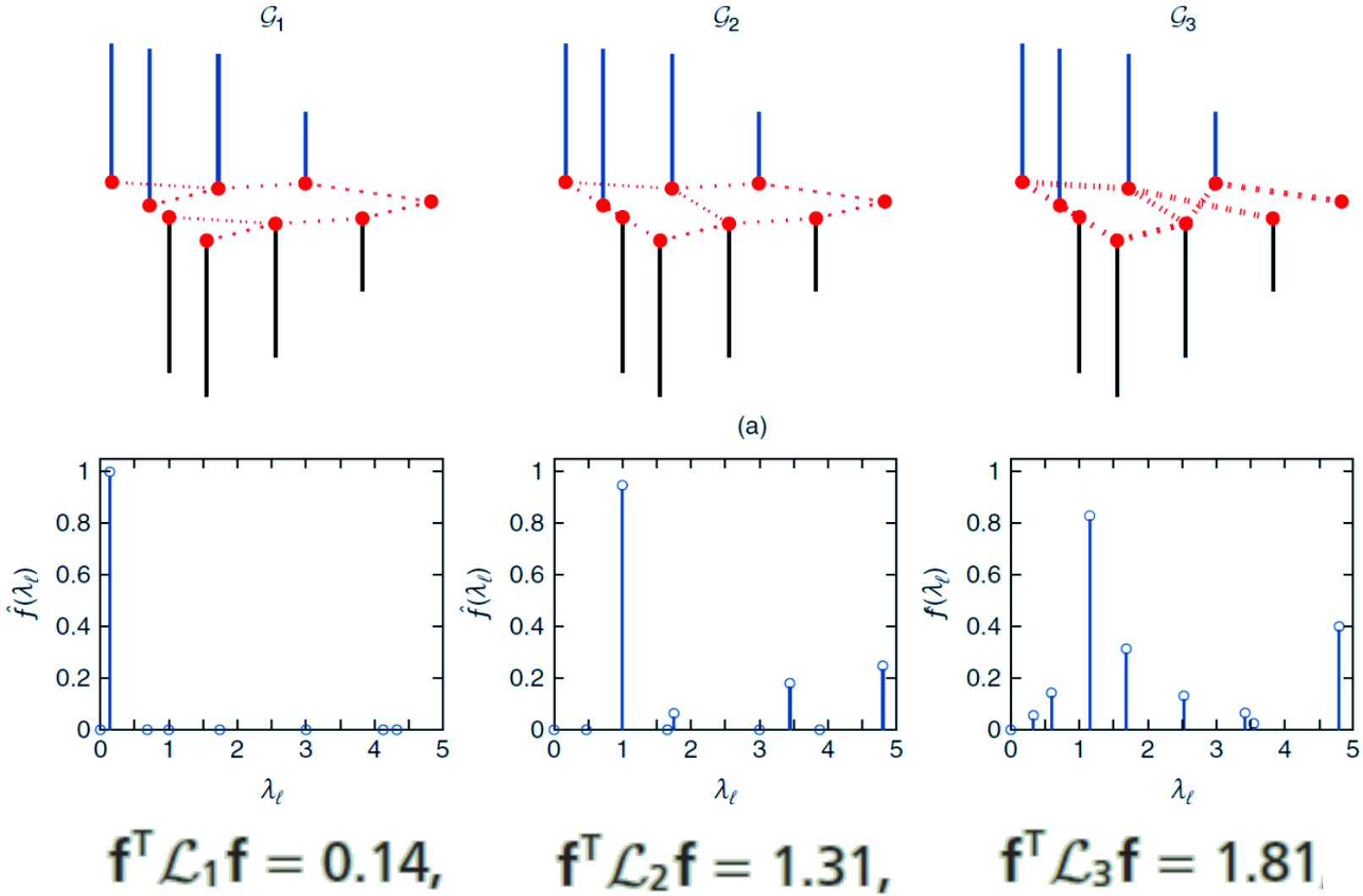
$$\downarrow$$

$$\operatorname*{argmin}_{\mathbf{f}} \{ \|\mathbf{f} - \mathbf{y}\|_2^2 + \gamma \mathbf{f}^{\mathrm{T}} \mathcal{L} \mathbf{f} \}.$$

Connectivity of the graph -> encoded in graph Laplacian

Define both a graph Fourier transform (graph Laplacian eigenvectors)

Different notions of smoothness


$\mathcal{G}_1$
$\mathcal{G}_2$
$\mathcal{G}_3$
(a)
$\hat{f}(\lambda_\ell)$
λ_ℓ
$\mathbf{f}^T\mathcal{L}_1\mathbf{f} = 0.14,$
$\mathbf{f}^T\mathcal{L}_2\mathbf{f} = 1.31,$
$\mathbf{f}^T\mathcal{L}_3\mathbf{f} = 1.81$

Total variation

- TV: Difference between two consecutive signal samples

$$\mathrm{TV}(\mathbf{s}) = \sum_n |s_n - s_{n-1}|.$$

- TV over graphs $\mathrm{TV}(\mathbf{s}) = \|\mathbf{s} - \mathbf{C}\mathbf{s}\|_1.$

Where C is the cyclic permutation matrix $\mathbf{A} = \mathbf{C} = \begin{bmatrix} & & & 1 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}$

- It measures of similarity between a graph signal and its shifted version
- Gradient $\frac{d\mathbf{s}}{dv_n} = \nabla_n(\mathbf{s}) = s_n - \sum_{m \in \mathcal{N}_n} \mathbf{A}^{\mathrm{norm}}_{n,m} s_m.$

$$\mathbf{A}^{\mathrm{norm}} = \frac{1}{\lambda_{\max}} \mathbf{A}.$$

Alternative view of TV

- The total variation of a vector $\mathbf{x}$ on a graph can also be defined as

$$TV_x : \|\mathbf{x}'\| = \left(\mathbf{x}^T \mathbf{L} \mathbf{x}\right)^{1/2}$$

- Substituting the inverse GFT $\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$, we get

$$TV_x^2 = \mathbf{x}^T \mathbf{L} \mathbf{x} = \hat{\mathbf{x}}^T \mathbf{\Lambda}\, \hat{\mathbf{x}} = \sum_{i=1}^{N} \lambda_i \|\hat{x}(i)\|^2$$

- Such that

$$|\hat{x}(i)| \leq \frac{TV_x}{\sqrt{\lambda_i}}$$

Filtering over graphs

The graph frequency λ_m is larger than graph frequency λ_ℓ if

$$\mathrm{TV}_{\mathrm{G}}\left(\mathbf{v}_m\right) > \mathrm{TV}_{\mathrm{G}}\left(\mathbf{v}_\ell\right).$$

Assuming the graph frequencies have been ordered from low to high, graph signal $\mathbf{s}$ is low-pass if its graph Fourier coefficients are zero for Ω_k, $k > \ell$, for some ℓ, $0 \leq \ell < N-1$.

We can similarly define band- and high-pass signals and filters.

Graph filters

- A graph filter is a system $H(\cdot)$ which takes a signal s and produces another signal $\hat{s} = H(s)$ at the output.

- The graph shift filter is a local operation that replaces a signal value s_n at node v_n with the linear combination of values at the neighbors of node v_n

$$\tilde{s}_n = \sum_{m \in \mathcal{N}_n} \mathbf{A}_{n,m} s_m.$$

- Hence, the output of the graph shift is given by the product of the input signal with the adjacency matrix of the graph:

$$\tilde{\mathbf{s}} = \begin{bmatrix} \tilde{s}_0 & \dots & \tilde{s}_{N-1} \end{bmatrix}^T = \mathbf{A}\mathbf{s}.$$

Graph filter

All linear shift-invariant graph filters in DSG
are polynomials in the adjacency matrix $\mathbf{A}$ of the form

$$h(\mathbf{A}) = h_0\mathbf{I} + h_1\mathbf{A} + \ldots + h_L\mathbf{A}^L$$

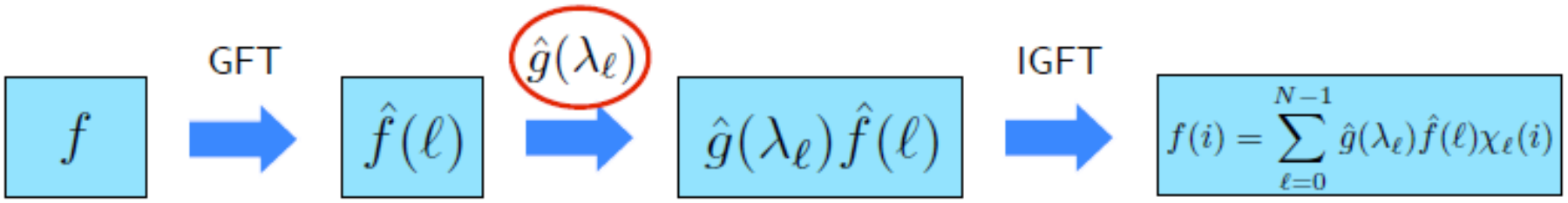
The output of the filter is the signal $\widetilde{s} = \mathbf{H}(s) = h(\mathbf{A})s$

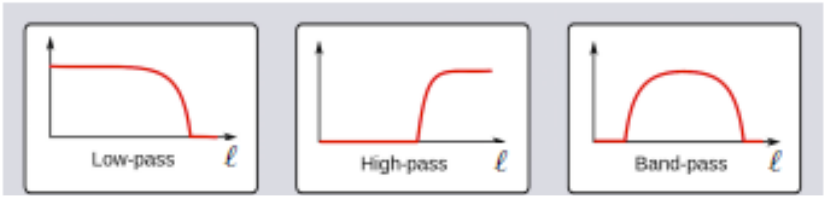
The GFT of a graph signal s is $\widehat{s} = \mathbf{F}s$,
where $\mathbf{F} = \mathbf{V}^{-1}$ is the graph Fourier transform matrix
obtained by $\mathbf{A} = \mathbf{VJV^{-1}}$

The values $\widehat{s}_n$ of the signal's graph Fourier transform
characterize the frequency content of the signal s.

Graph spectral filtering

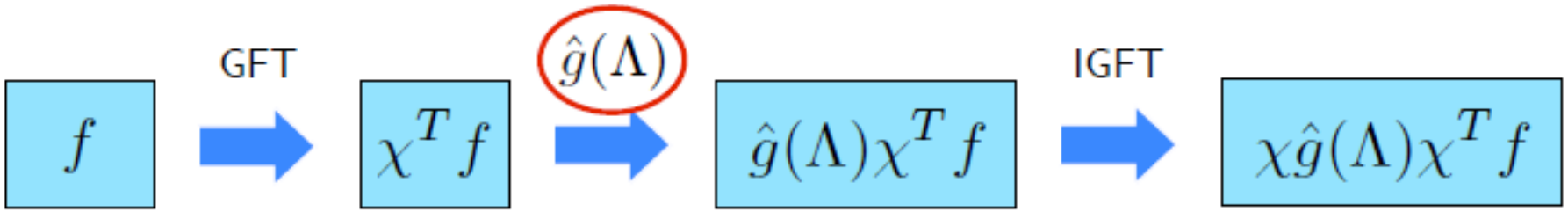
$$\text{GFT:}\quad \hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(i)$$





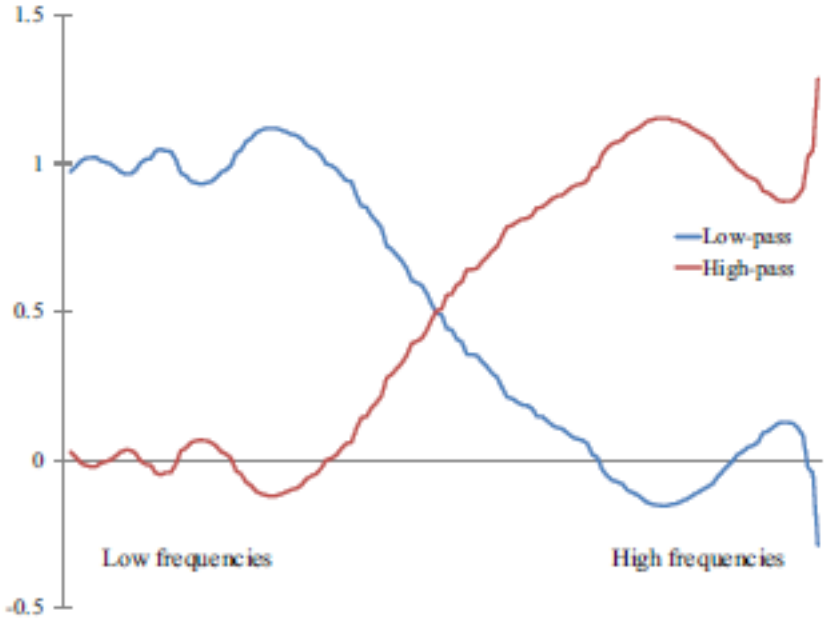
Graph spectral filtering

$$\text{GFT:}\quad \hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$$



$$\hat{g}(\Lambda) = \begin{bmatrix} \hat{g}(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \hat{g}(\lambda_{N-1}) \end{bmatrix}$$

Example


1.5
1
0.5
0
-0.5
Low-pass
High-pass
Low frequencies
High frequencies

Denoising signals over graphs

- Let **t=x+w** be the measured signal where **x** is the true signal and **w** is the noise.
- Impose graph signal smoothness in quadratic form

$$\mathrm{S}_2(\mathbf{x}) = \frac{1}{2}\,||\mathbf{x} - \mathrm{A}\,\mathbf{x}||_2^2$$

- Denoising $\quad \tilde{\mathbf{x}} = \mathbf{argmin}\ \frac{1}{2}||\mathbf{x}-\mathbf{t}||_2^2 + \alpha\,\mathrm{S}_2(\mathbf{x}).$
- Gradient of objective

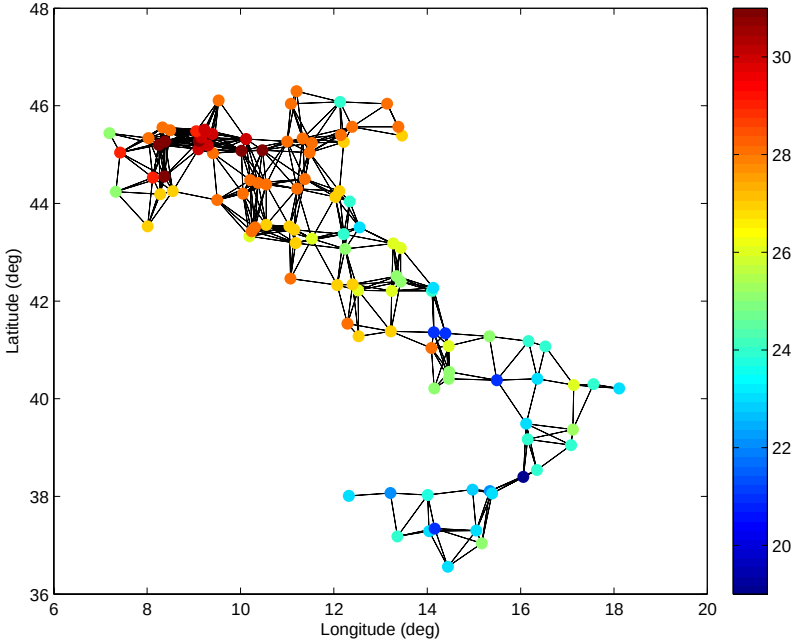
$$\begin{aligned} & \frac{\partial}{\partial \mathbf{x}}\left(\frac{1}{2}||\mathbf{x}-\mathbf{t}||_2^2 + \alpha\,\mathrm{S}_2(\mathbf{x})\right) \\ = \ & \frac{1}{2}\frac{\partial}{\partial \mathbf{x}}\left((\mathbf{x}-\mathbf{t})^*(\mathbf{x}-\mathbf{t}) + \alpha\,\mathbf{x}^*(\mathrm{I}-\mathrm{A})^*(\mathrm{I}-\mathrm{A})\,\mathbf{x}\right) \\ = \ & (\mathbf{x}-\mathbf{t}) + \alpha(\mathrm{I}-\mathrm{A})^*(\mathrm{I}-\mathrm{A})\,\mathbf{x}, \end{aligned}$$

- Solution

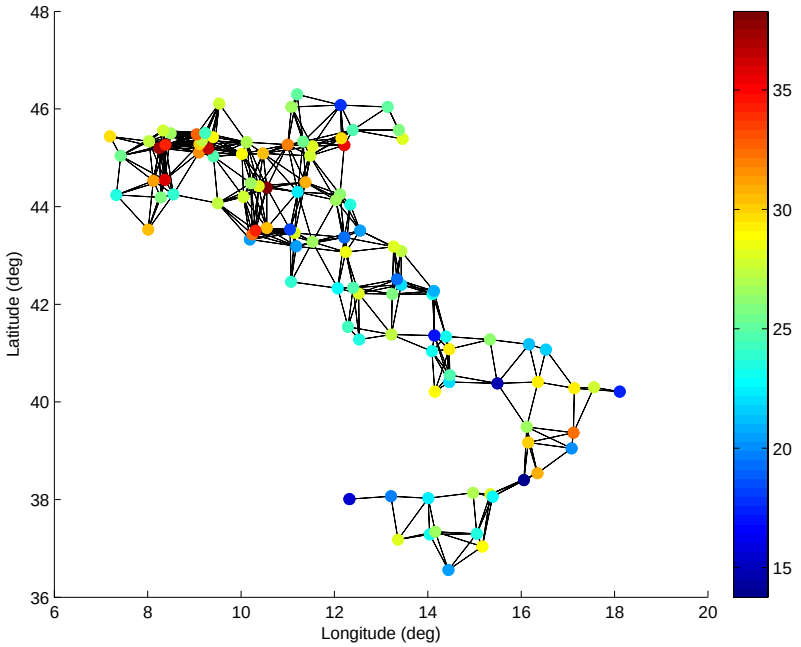
$$\tilde{\mathbf{x}} = \left(\mathrm{I} + \alpha(\mathrm{I}-\mathrm{A})^*(\mathrm{I}-\mathrm{A})\right)^{-1}\mathbf{t}.$$

Denoising example

Temperature field

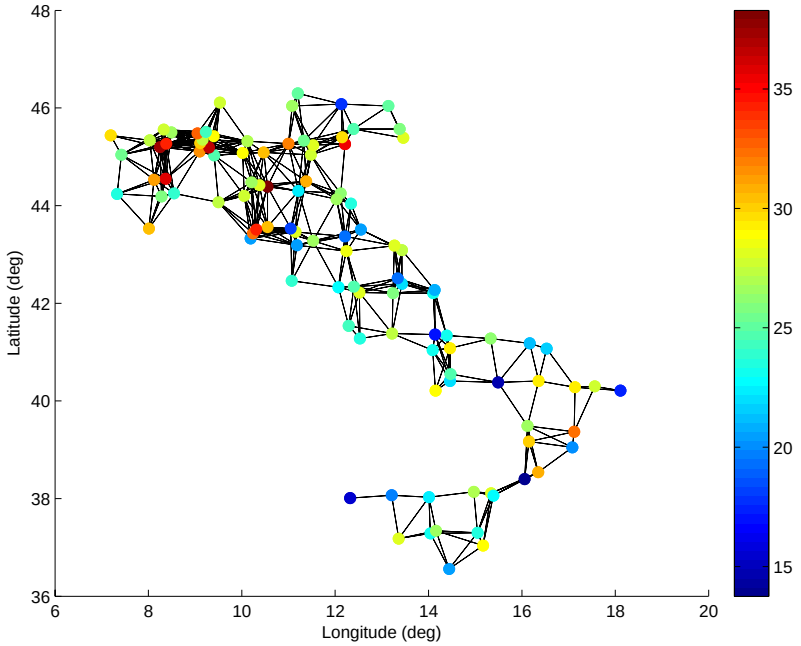


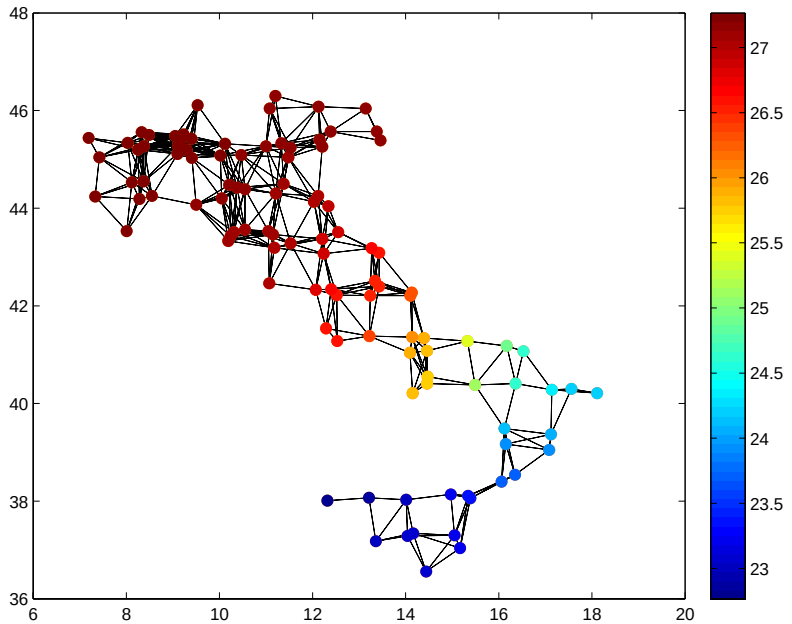
Noisy signal



Denoising example

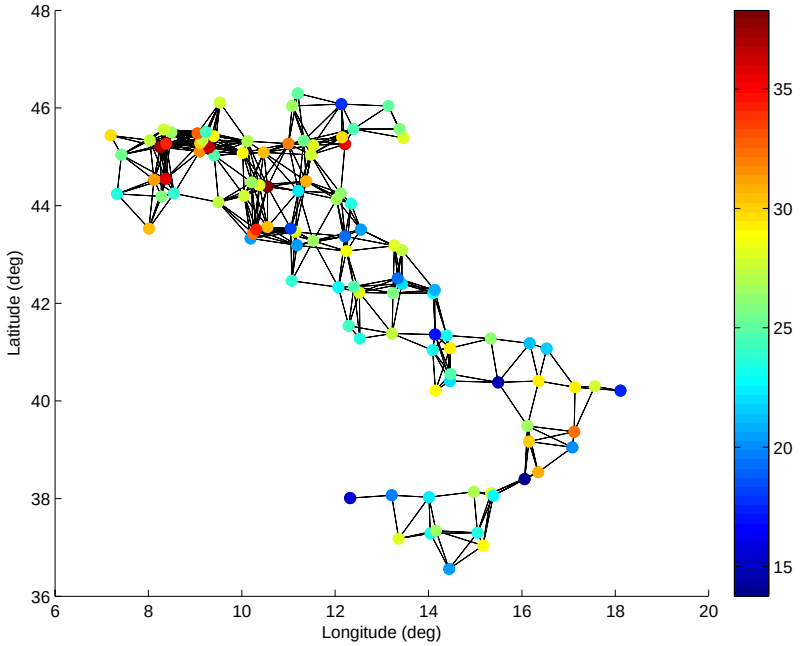
Reconstruction with 2 eigenvectors



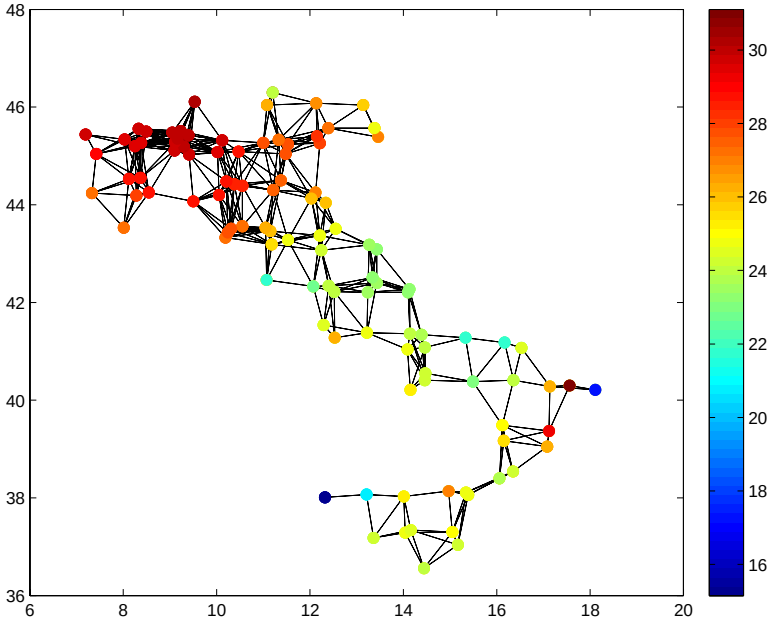


Denoising example

Noisy temp. field



Reconstruction with 24 eigenvectors



Spatial Signal Graphs

1-hop averaging transform

$$y[n] = \frac{1}{d_n} \sum_{m=1}^{N} A[n, m] x[m]$$

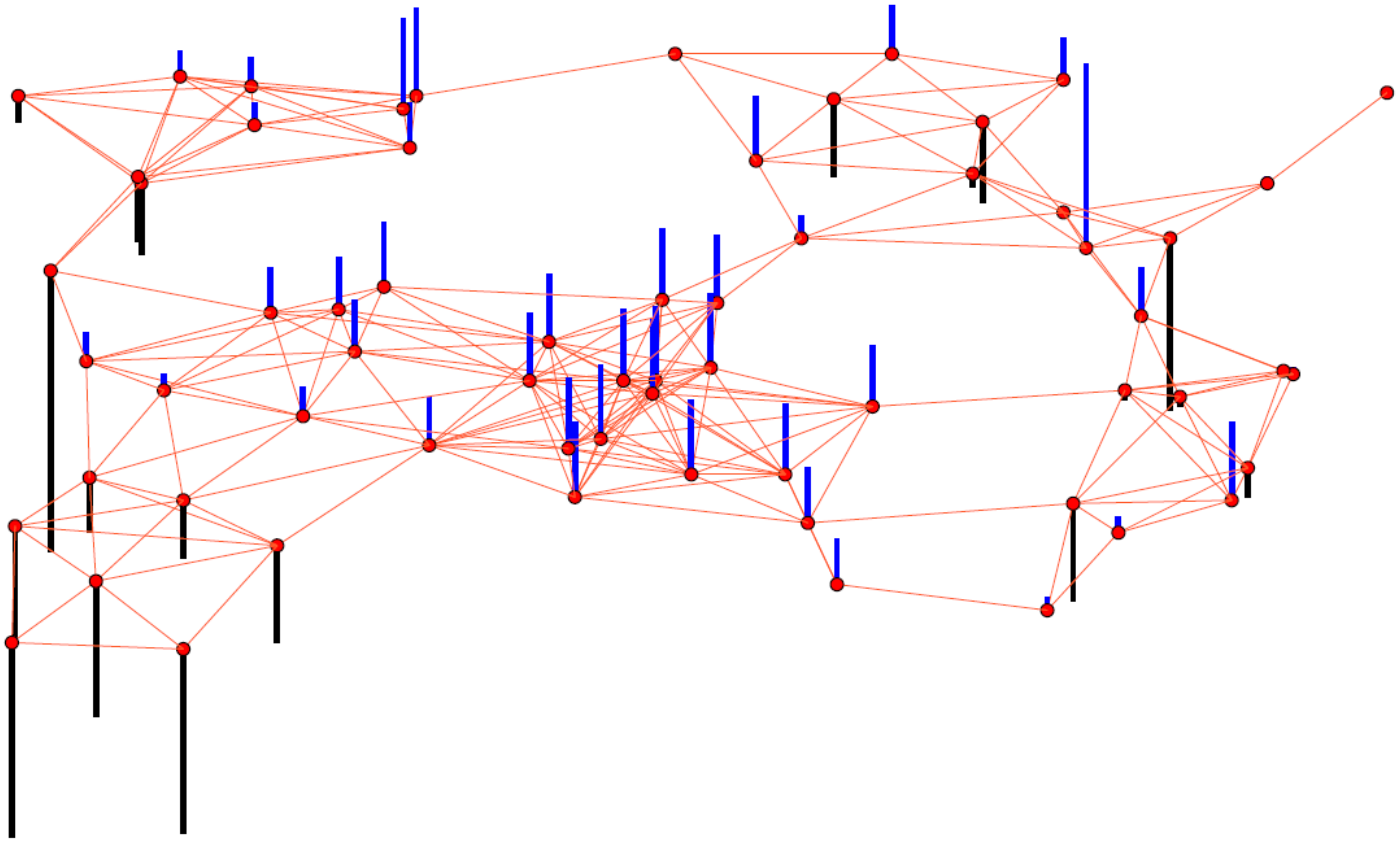
1-hop difference transform

$$y[n] = \frac{1}{d_n} \sum_{m=1}^{N} A[n, m](x[n] - x[m])$$

$$\mathbf{y} = \mathbf{D}^{-1}\mathbf{A}\mathbf{x} = \mathbf{P}_{rw}\mathbf{x}$$

$$\mathbf{y} = \mathcal{L}_{\mathbf{rw}}\mathbf{x} = \mathbf{x} - \mathbf{P}_{\mathbf{rw}}\mathbf{x}$$

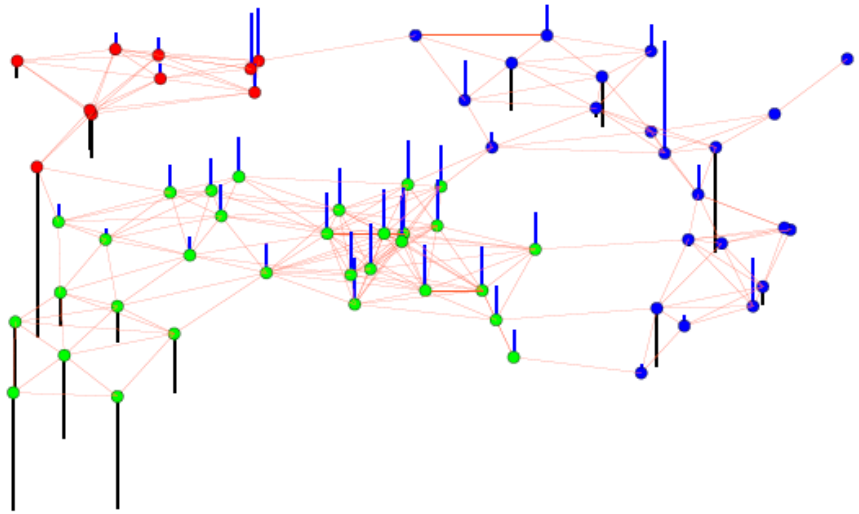
Find structure in data

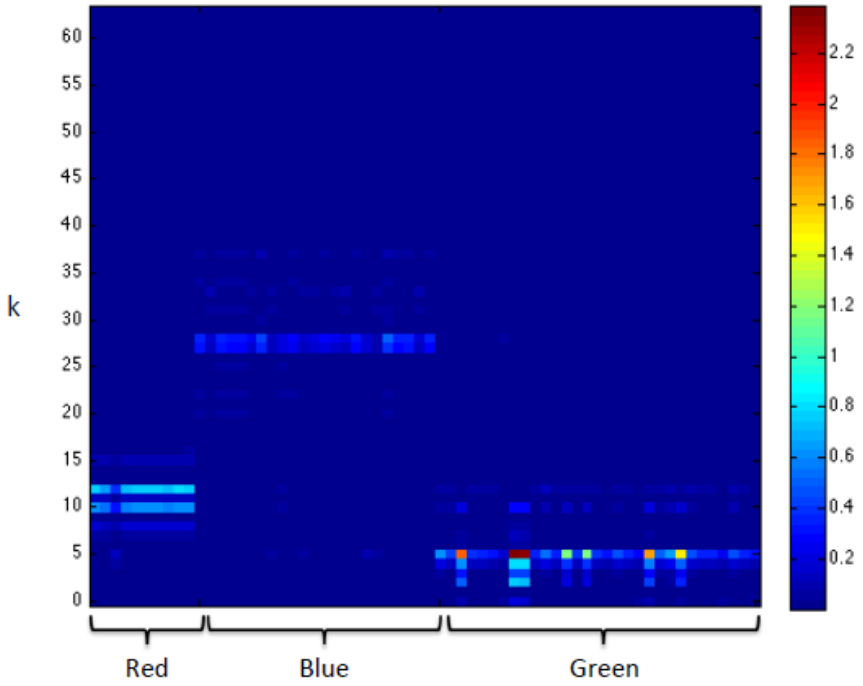


Any structure?

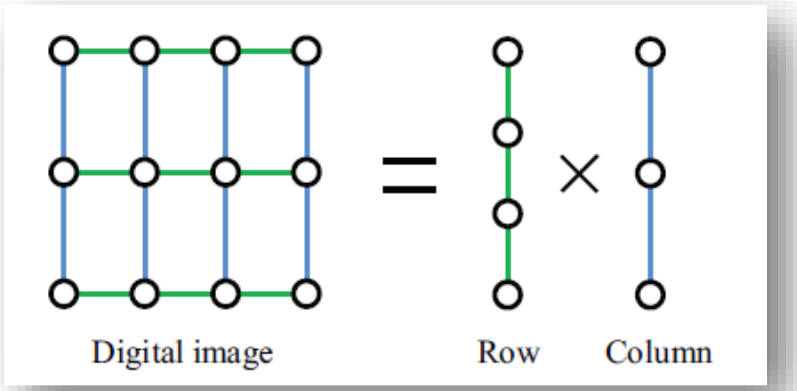
Find structure in data

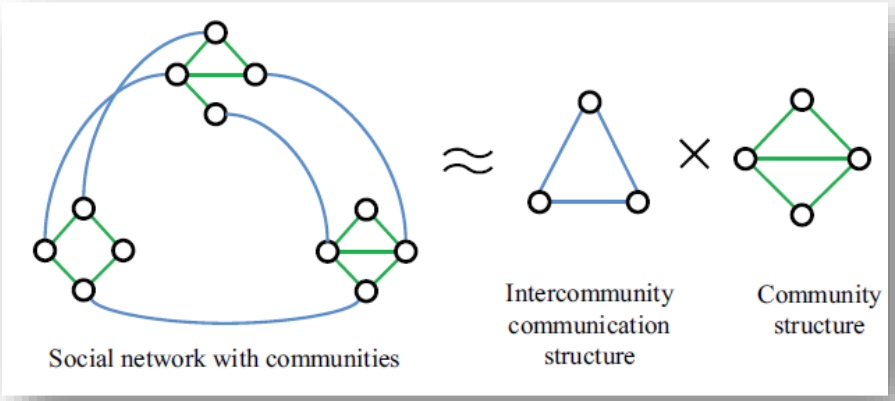
- Signal f comprised of three different graph Laplacian eigenvectors (u_{10}, u_{27}, u_5) restricted to the three different clusters of vertices

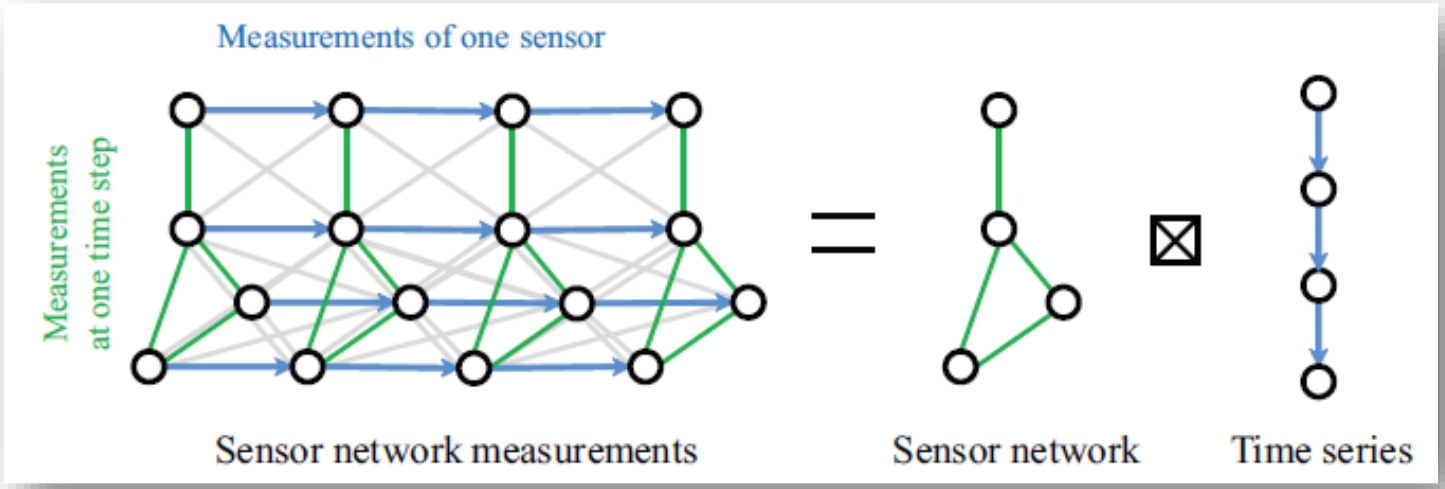




Product graphs







Spectral anomaly detection in WSN

Decomposition of Laplacian $\mathcal{L}_G = \mathbf{U}_G \mathbf{\Lambda}_G \mathbf{U}_G^t$

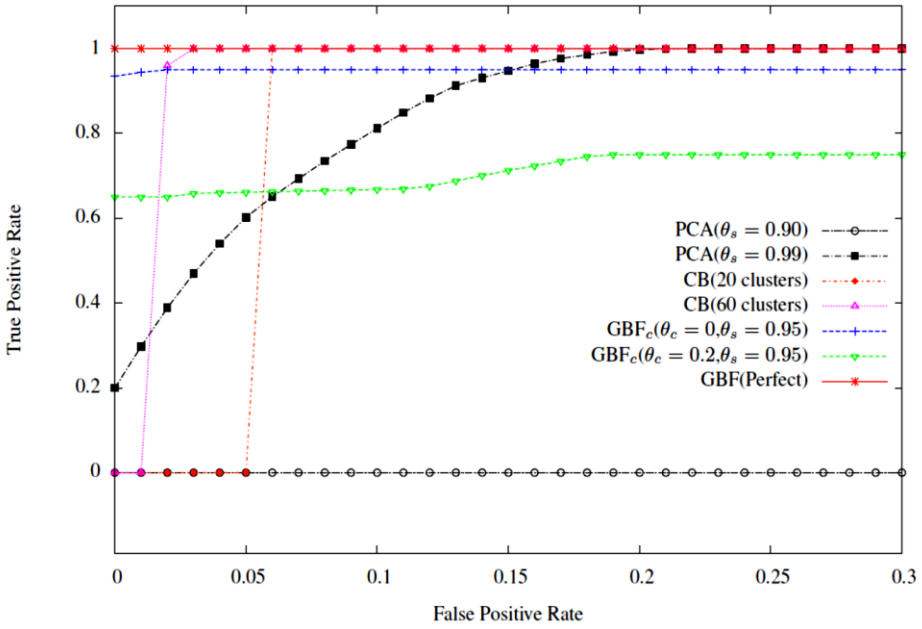
Alternative approach $h(\mathcal{L}_G) = \mathbf{U}_G (h(\mathbf{\Lambda}_G)) \mathbf{U}_G^t$

Graph construction $[w_{i,j}]_b = \exp\left(-\frac{(1-\|\rho(i,j)\|_1)^2}{\Delta_c^2}\right) \cdot \exp\left(-\frac{\tilde{D}(i,j)^2}{\Delta_d^2}\right)$

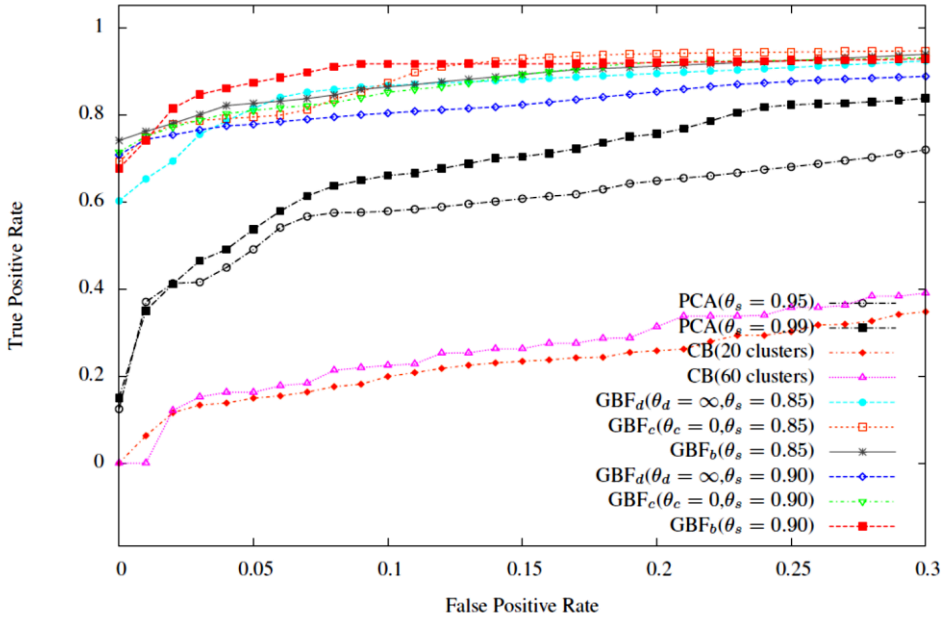
Data fit in graph $\sigma_l^2 = s^2[\mathbf{u}_l^t \mathbf{X}]$ where $s^2[\mathbf{p}^t]$ is the sample variance

Target ratio $\arg\max_c \sum_{l=1}^{c} \frac{\sigma_l^2}{\sigma_T^2}$ subject to $\sum_{l=1}^{c} \frac{\sigma_l^2}{\sigma_T^2} \leq \theta_s$

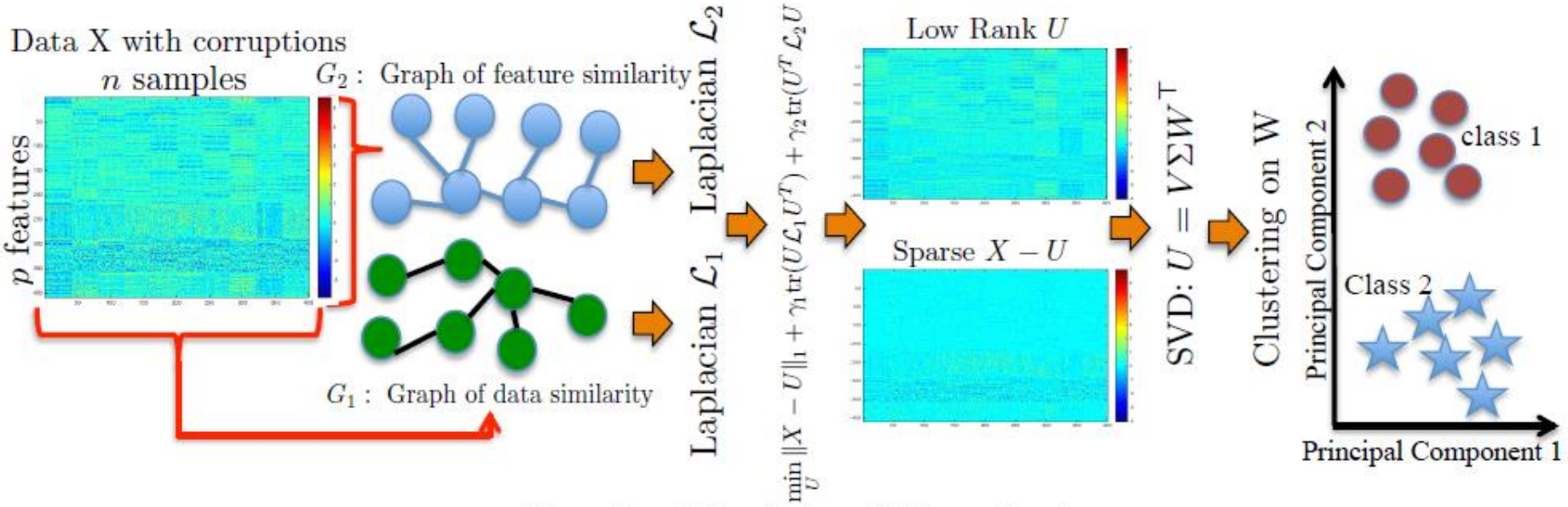
Global



Distributed



Robust PCA on graphs



Main idea of Fast Robust PCA on Graphs

$$\min_{U} \|X - U\|_1 + \gamma_1 \operatorname{tr}(U\,\mathcal{L}_1\,U^\top) + \gamma_2 \operatorname{tr}(U^\top \mathcal{L}_2\,U).$$

$$\min_{U,S} \|S\|_1 + \gamma_1 \operatorname{tr}(U\,\mathcal{L}_1\,U^\top) + \gamma_2 \operatorname{tr}(U^\top \mathcal{L}_2\,U),$$
$$\text{s.t. } X = U + S,$$

Background Separation from Videos via PCA

Reading material

- Ortega, Antonio, et al. "Graph signal processing: Overview, challenges, and applications." Proceedings of the IEEE 106.5 (2018): 808-828.
- Shuman, David I., et al. "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains." arXiv preprint arXiv:1211.0053 (2012).
- Sandryhaila, Aliaksei, and José MF Moura. "Discrete signal processing on graphs." IEEE transactions on signal processing 61.7 (2013): 1644-1656.